SAGA API for gLite Service Discovery

Steve Fisher and A Paventhan

3rd EGEE Users Forum, 11-14 Feb 2008

Le Polydôme, Clermont-Ferrand, France

www.eu-egee.org

Information Society and Media

Overview

- Problem
- Current solutions
- SAGA
- The SAGA SD API
- Status
- Summary

The Problem

- **Middleware services**
  - are dependent on using other middleware services and need to make selections based on various characteristics of those services
- **Users**
  - especially those arriving with their laptop at a friendly site may also need to locate services.

Various Solutions – for services

- **Configuration files**
  - This does not reflect the current state of the service
  - The configuration file maybe out of date
    - The service no longer exists
    - There are new services not in the configuration file

- **Specialised directory services**
  - UDDI
  - Grimoires

- **Information system queries**
  - These are tied to a specific implementation – LDAP, R-GMA
  - These depend upon the precise model of a service (GLUE n)

- **API**
  - May be on top of any of the above

Various Solutions – for users

- If user is not mobile then same set of solutions apply
- However if he moves around with laptop then he also needs a bootstrap mechanism
  - Bonjour (from Apple)
  - These mechanisms rely upon multicast and so are only easily applied on a small scale
  - Won’t be mentioned further in this talk

gLite SD API

- This was the gLite solution worked out between the Data Management people and JRA1-UK
- It lacks some features and to extend it would make it rather complex – by addition of more functions

SAGA – OGF Groups

- **The Simple API for Grid Applications (SAGA) from the Open Grid Forum (OGF) defines standard APIs to allow grid applications to be middleware independent**

- **The specification of services, and the protocols to interact with them, is out of scope**

- **Main specification includes:**
  - Job, File and Replica Management
  - Name Spaces, Streams and RPC

- **Extra specifications as extensions**

- **SAGA components have a "plugin" architecture**
  - Various adapters exist for Globus and OMII

- **We have contributed a Service Discovery specification to SAGA and an implementation as a generalisation of the existing gLite component**

SAGA API for Service Discovery

- **Dynamically discover working services as and when required**
- **Implemented as a generalisation of the existing gLite API**
- **Supports a selection mechanism built on the GLUE service model**
  - Compatible with the new evolving GLUE specification
  - Search criteria are specified by means of SQL style filters:
    - Service
    - VO
    - Data
- **Plugins/adapters required for each underlying information system**

Filters

- The service filter selects on the basis of some GLUE attributes such as the “type” of service
  - string svc_filter = "type = 'Broker' AND
    name = 'CERN-PROD-rb‘";

- The VO filter allows the user to select from those services available to specific VOs
  - string vo_filter = "VO IN ('atlas', 'dteam')";

- The data filter makes use of a GLUE feature of key/value pairs associated with each service - this makes it extensible
  - string data_filter = "RunningJobs > 10";

Example

```
saga::discoverer d (SAGA_DEFAULT_SESSION);
vector<saga::service_description> slist =
          d.list_services(svc_filter, vo_filter, data_filter);
if  (slist.size() > 0) {
     string url = slist[0].get_url();
} else {
    cerr << “No suitable service found” << endl;
}
```

- Say what type of service you want
- Use one of them - could be first or choose at random

Example – for generating a web page of services

```
saga::discoverer d (SAGA_DEFAULT_SESSION);

vector<saga::service_description> slist =
        d.list_services(svc_filter, vo_filter, data_filter);

for (int i = 0; i < slist.size(); i++) {
     attrib_names = slist[i].list_attributes();
     cout << "SERVICE #" << i <<  endl;
     for (int j = 0; j < attrib_names.size(); j++) {
         cout << attrib_names[j]
                  << "="
                  << slist[i].get_attribute(attrib_names[j])
                  << endl;
     }
}
```

Status

- **Currently**
  - Specification is about to go to the OGF editor
  - SD Interface (C++) is being added to the SAGA repository (gLite external)
  - R-GMA and BDII adapter in C++ in gLite CVS

- **Future Development**
  - Python wrapper
  - C wrapper
  - Java implementation

  - The wrappers avoid the need for rewriting the plugins
  - JNI will be considered for a Java wrapper as a short term solution

Summary

- **The SAGA approach in general is useful from a user perspective because it frees end users from dependency upon specific Grid middleware**

- **We expect that the use of SAGA will grow making the interface increasingly valuable**

- **The SAGA Service Discovery API has been designed to be very easy to use with SQL style filter expressions**
  - It also means that services that use other services require less error-prone configuration as they can find the actual services when they need them
  - This in turn leads to increased reliability
  - Spec about to go to OGF editor

- **The SAGA based Service Discovery C++ API supporting SQL style filters has been implemented within gLite**

With thanks to:

Information Society and Media

GridPP UK Computing for Particle Physics